IMPLEMENTATION OF ANISOTROPIC FLOW INTO THE TOUGH2 CODE

[1]Andri Arnaldsson, [1]Jean-Claude Berthet, [1]Snorri Kjaran, [2]Sven Þ. Sigurðsson

[1]Vatnaskil Consulting Engineers
Suðurlandsbraut 50
108 Reykjavík, Iceland
[2]School of Engineering and Sciences, University of Iceland
Hjarðarhagi 7
107 Reykjavík, Iceland
e-mail: andri@vatnaskil.is

ABSTRACT

Characteristics of groundwater flow in porous media can be heavily influenced by local anisotropy in intrinsic permeability. Therefore, the ability of a numerical model to properly simulate anisotropy is vital in designing a well-calibrated model capable of predicting the response of a system under stress. A new numerical scheme for fully tensorial treatment of anisotropic flow within model layers (2D) has been designed and implemented into the TOUGH family of simulators. Previously, the TOUGH simulators were only able to accurately resolve anisotropic flow in two cases, i.e., when the connections between two gridblocks were either along the principal direction or along a streamline.

A simulation using the new anisotropic scheme is activated by a MOP parameter and the existence of two new data blocks in the TOUGH input file. The first block (ANISO) contains the gridblock name, coordinates, permeability values along the first and second principal axes, and the angle of the principal direction from the local x-axis. The second block (SEGST) lists the gridblocks for each connection needed to calculate the additional flow gradients in the Jacobian matrix, along with a vector along the interface between the neighboring gridblocks.

Testing has shown that the new scheme adds approximately 16% to the execution time of multiphase (water and steam) simulations using the TOUGH2 simulator. In order to account for anisotropy between two gridblocks, flow from neighboring gridblocks must be incorporated into the calculations as well, resulting in more off-diagonal elements in the Jacobian matrix. About 33% of the execution time is used for constructing the Jacobian matrix elements, and another 25% is used for sorting the array containing the matrix elements during each iteration.

The new scheme has been rigorously tested against simple theoretical solutions to Darcy's law, as well as more complicated examples solved by numerical software packages with anisotropic flow capabilities. In all cases, a good agreement with the new scheme has been found.

This work has been performed as a part of a MoU (Memorandum of Understanding) between the Earth Sciences Division of Lawrence Berkeley National Laboratory and Vatnaskil Consulting Engineers.

INTRODUCTION

Here, we present a short description of the numerical scheme that has been implemented into the TOUGH2 simulator. For simplification, the emphasis will be on horizontal two-dimensional flow.

Consider a general mesh as shown in Figure 1. The goal is to calculate the flow through the face *st* connecting elements (gridblocks) O and B by including full permeability anisotropy. The mesh was generated with a Voronoi tessellation (de Berg, 2000) which guarantees that the line connecting the interior "central" points O and B will be perpendicular to the line *st* and will be exactly halved by that line. Note, on the other hand, that the line segment *st* will in general not be halved by the line OB, and the line OB may indeed intersect *st* at a point exterior to the segment *st*.

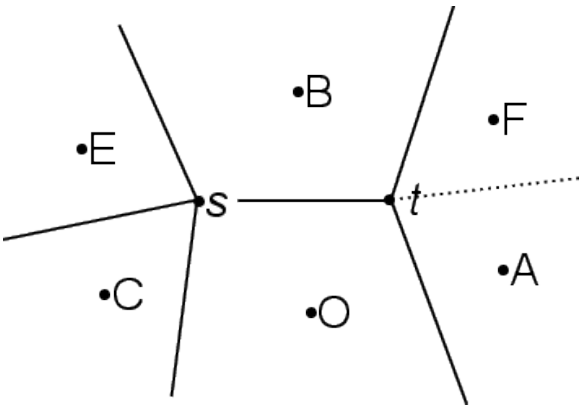

Figure 1. A general Voronoi mesh. The aim is to calculate the flow through the line segment *st* connecting points O and B.

The figure defines the points O, B, C, E, A, F, which are the centers of elements surrounding the edge (interface area) in question. Points *s* and *t* are defined as the vertices of the edge. Elements C and E will always be in contact with edge *st* through vertex *s*, and elements A and F will always be in contact through vertex *t*. Elements A and C will always be the elements sharing an edge with Element O. Elements E and F will always be elements sharing an edge with Element B. When one vertex of the edge *st* is on an outer boundary of the region, it will always be vertex *t*, and thus vertex *s* will be inside the region. Moreover, in such cases, elements A and F will not exist.

Within our implementation, all information pertaining to the elements are stored in arrays in the following order: O, B, C, E, A, F. The inclusion of the additional points, A, C, E, and F, serves to obtain an approximation of the gradient for the pressure along the edge *st*, which in turn allows us to take into account the anisotropy when calculating the flow across the edge. This approximation of the gradient is a linear combination of the approximate gradients in the triangular planes OBE, OBF, BOA, and BOC. Following this setup, the flow through *st* can be compactly represented as (Sigurðsson, 2009)

$$\mathrm{F} = \frac{k_r \rho}{\mu}\left[\alpha_O\ \alpha_B\ \alpha_C\ \alpha_E\ \alpha_A\ \alpha_F\right]\begin{bmatrix} P_O \\ P_B \\ P_C \\ P_E \\ P_A \\ P_F \end{bmatrix} \qquad (1)$$

with k_r, ρ, and μ as the relative permeability, density, and viscosity, respectively. The *P* vector represents pressure at each element center. As an example, an element in the vector α is of the form

$$\alpha_O = \frac{h}{4}\left[s_y - t_y \quad s_x - t_x\right]\begin{bmatrix} k_x & k_{xy} \\ k_{xy} & k_y \end{bmatrix} \left[\begin{array}{l} \frac{1}{2\lvert OCB\rvert}\begin{bmatrix} C_y - B_y \\ B_x - C_x \end{bmatrix} + \frac{1}{2\lvert OAB\rvert}\begin{bmatrix} B_y - A_y \\ A_x - B_x \end{bmatrix} + \\ \frac{1}{2\lvert OEB\rvert}\begin{bmatrix} E_y - B_y \\ B_x - E_x \end{bmatrix} + \frac{1}{2\lvert OFB\rvert}\begin{bmatrix} B_y - F_y \\ F_x - B_x \end{bmatrix} \end{array}\right] \qquad (2)$$

Here, h is the model or layer thickness and k_x, k_y, k_{xy}, the elements of the permeability tensor, are:

$$k_x = k_n \cos^2\theta + k_t \sin^2\theta \qquad (3)$$

$$k_y = k_n \sin^2\theta + k_t \cos^2\theta \qquad (4)$$

$$k_{xy} = (k_n - k_t)\sin\theta\cos\theta \qquad (5)$$

where k_n and k_t (Strack, 1989) are the permeability values along the principal directions and θ is the angle of the first principal direction from the x-axis. At the boundaries, the α values are calculated in a different manner. With vertex *t* being on the boundary (see Figure 1), triangles OAB and OFB are missing. Therefore, the contributions from triangles OCB and OEB are doubled. When the vertices are in contact with only three elements, the C term is equal to E and A is equal to F.

CHANGES TO THE CODE

Subroutine INPUT

Two sections have been added to subroutine INPUT: ANISO (label 3200) and SEGST (label 3300). New common blocks have been added to the program to store the extra data required by the anisotropic method, including the element coordinates, the permeability tensors, and the interface area vectors *st*. When the anisotropy method is used, up to six elements may contribute to the flow along a connection, as opposed to only two elements when the isotropic method is used. New common blocks have been added to subroutine INPUT and to the program to store data for the extra elements. Note that for a connection, the elements in arrays ELEM1, ... and NEX1, ... must correspond, respectively, to elements O, B, C, E, A, F. When close to a boundary, ELEM5 and ELEM6 shall be left

blank, and NEX5 and NEX6 will be set to zero by the program.

Two new sections were added to the subroutine to parse the new input blocks, ANISO and SEGST. The format of the new blocks is discussed later in this article.

Subroutine ANISOTROPY

Subroutine ANISOTROPY is triggered when a block ANISO is detected in the input. It is possible, though, to force the run in either isotropic or anisotropic mode by means of a MOP parameter. The new subroutine is called from subroutine MULTI to supplant the loop over connections implemented there. The loop has been implemented again in the new subroutine, ANISOTROPY, and has a structure similar to that of the original code. It includes a flux loop to calculate the finite differences required to build the Jacobian matrix of the residuals (Pruess, 1999). Including anisotropy introduces additional non-zero elements into the Jacobian matrix. For instance, in the isotropic case, the Jacobian element for the neighboring elements B and C is zero. In the anisotropic case, it is no longer zero, since the flow between elements O and B depends on the pressure in Element C. Note, however, that these additional elements in the Jacobian only depend on physical parameters in the regions surrounding points O and B, since the additional points only serve to obtain an approximation of the gradient of the pressure along the edge *st*. Therefore, no extra finite difference is required to compute the new Jacobian elements, since the flow across segment *st* is a linear function of the pressure in elements C, E, F, A (i.e., in Equation 1, the permeability, viscosity, and density are functions of P_O and P_B but not of P_C, P_E, P_F, and P_A).

ADDITIONAL INPUT

Two extra data blocks are required to run a simulation using anisotropy. Throughout this work, AMESH (Haukwa, 1998) was used to generate the necessary input for TOUGH2. An external program was then used to convert the output data produced by AMESH into input data needed for anisotropy.

Block ANISO

Block ANISO has the following structure:

- element names (columns 1 to 5),
- element *x*, *y,* and *z* coordinates (6 to 20, 21 to 35, 36 to 50),
- permeability along first and second principal axes (51 to 60, 61 to 70),
- angle between first principal axis and axis *x* (71 to 80).

Block ANISO must always come after block ELEME, and the elements must be in the same order.

Block SEGST

Block SEGST has the following input:

- names of elements O, B, C, E, A, F (columns 1 to 30, five characters each),
- *x* and *y* coordinates of vector *st* (31 to 45, 46 to 60).

Block SEGST must come after block CONNE, and the connections must be in the same order. A line with four elements (A and F blank) indicates that the connection is next to a boundary. A line with two elements means that the connection is parallel to the *z* axis, which will always be treated as the vertical (as well as a principal) axis. Note that in such a case, the vector *st* is not used by the program. An interface area vector *st* must never be zero.

Care must be taken that block SEGST is consistent with the data provided in block CONNE. Connections in the *xy* plane must have their ISOT number set to 1 or 2. These connections will be treated anisotropically using the permeability provided in input block ANISO. Connections parallel to *z* must have their ISOT number set to 3. Those connections will be treated isotropically using the permeability provided in input block ROCKS. BETAX, the cosine of the angle between the connection and the gravity axis, must be zero for the *xy* connections (ISOT 1 or 2), and 1 or -1 for the *z* connection (ISOT 3). Parameters PER(1) and PER(2) in block ROCKS are not used in the simulation, since the permeability values for connections in the *xy* plane are defined in the ANISO block.

PROFILING AND PERFORMANCE

A test performed on a hypothetical multiphase system reveals that the anisotropic method is about 16% slower than the original isotropic method. About 25% of the computation time is used in sorting the array containing the Jacobian elements (array CO). Since array CO is always filled in the same disordered manner, computation time could be reduced by implementing a system that memorizes the positions of the elements after sorting. This would require the array to be ordered only once at the beginning of the simulation.

In the original isotropic implementation of TOUGH2, array CO is sorted inside the solver. Our anisotropic implementation requires the array to be sorted earlier. To prevent the solver from attempting to sort the array again, a Boolean parameter SORTED was added to the solver subroutines. The parameter is used to inform the solver that it is unnecessary to sort the array again. This new feature reduces computation time by several percentage points.

About 33% of the computation time is spent in subroutine ANISOTROPY, which contains the essential lines of code needed to implement anisotropy, excluding the sorting algorithm. Averaging of the absolute permeability is performed according to the definitions in MOP(11).

EXAMPLES

Four examples are presented in this section. Each one validates the correct implementation, or re-implementation when necessary, of different sections and/or features of the program. To begin, we shall discuss the comparison with the Theis solution in order to verify our approach.

Comparison with Analytical Theis Solution

The Theis solution gives the drawdown in an infinite homogeneous aquifer under constant pumping (Bear, 1979). It is represented here in the form of pressure change as a function of several parameters that are either TOUGH2 input parameters or parameters generated by the equation of state module (EOS1) in TOUGH2:

$$\Delta P = \frac{Q}{4\pi T\beta} W\left(\frac{S}{4Tt}\left(u_1^2 + \frac{1}{\beta^2}u_2^2\right)\right) \qquad (5)$$

Here, T is the aquifer transmissivity (m^2/s), S is the aquifer storage coefficient, Q is the production rate (kg/s), t is the time (s) and u_1 and u_2 are the coordinates in the principal coordinate system. When the coordinates differ from the canonical x and y system, the coordinates must be rotated according to:

$$u_1 = x\cos\theta + y\sin\theta \qquad (6)$$
$$u_2 = -x\sin\theta + y\cos\theta \qquad (7)$$

with θ as the angle between u_1 and x. The well function, W, is in this case the exponential integral (Abramowitz et al., 1965)

$$W(x) = \int_{-\infty}^{x} \frac{e^t}{t} dt \qquad (8)$$

A test was performed using a simple one-layer square mesh of size 5000×5000 m^2 consisting of elements of size 50×50 m^2. Using a mesh consisting of regular hexagonal elements with 80 m side length provided identical results. A sink was attached to an element close to the center, producing 10 kg/s of water. The model consisted of an anisotropic rock having permeability values along its principal axes of 10^{-14} and 10^{-15} m^2. The angle between the first axis and the x axis was set to 0.2π rad. The porosity was set to 0.1. Initial pressure was set to 50 bar and initial temperature to 220°C for all elements.

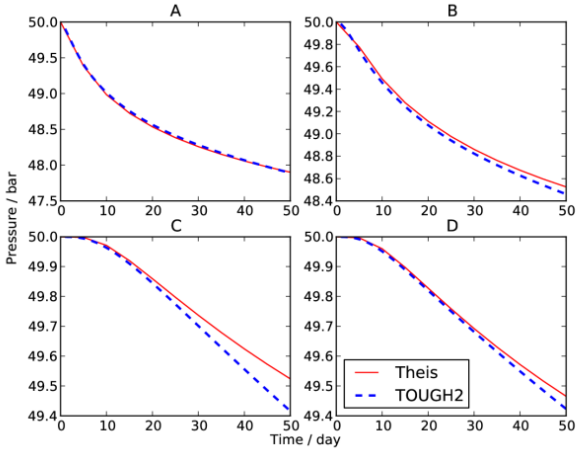

Figure 2. Comparison between TOUGH2 (including anisotropy) and the Theis solution for anisotropic aquifer at four different points.

The pressure was recorded at four different points within the model during a run of 50 days. The results are shown in Figure 2. A contour plot of the simulated pressure across the model area at the end of the 50 days is shown in Figure 3. The location of the four different calibration points is also shown in Figure 3.

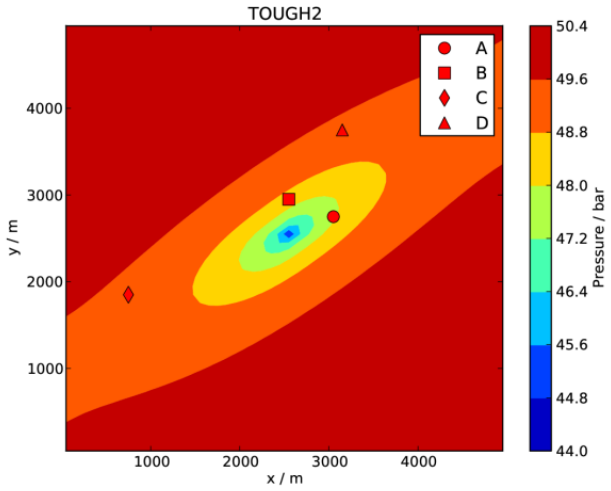

Figure 3. Areal plot of simulated pressure contours generated by TOUGH2 after 50 days. The red symbols show the locations of the four calibration points represented in Figure 2.

The results of the TOUGH2 simulation plotted in Figure 2 show good agreement with the Theis solution. As the simulation time progresses, the curves begin to diverge due to the no-flow boundaries used in the TOUGH2 model. The Theis solution assumes an aquifer of infinite extent. The discrepancy becomes more prominent closer to the boundary as time passes.

The comparison with the Theis solution confirms the correct behavior of the method and implementation presented here.

Immobile Water

In this and all subsequent examples, the calculations were performed on a rectangular mesh grid with 50 m between two adjacent points and a total area of 4750×4750 m^2.

Here, the new implementation is compared to the original TOUGH2 code by setting the permeability values to $k_x = 10^{-13}$ m^2, $k_y = 10^{-11}$ m^2 and rotating the principal axes along x and y axes to allow for direct comparison. The following example represents a two-phase system in which the water phase is immobile due to a low saturation level (below the residual saturation) (Grant, 1982). A producing well located at $(x,y) = (0,0)$ m produced at a constant rate of 30 kg/s. All boundaries are assumed to be impermeable (no flow).

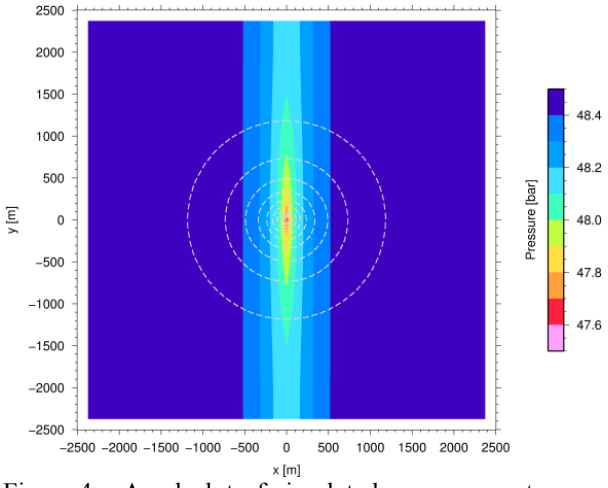

Figure 4. Areal plot of simulated pressure contours generated by TOUGH2 after five days of producing at a constant rate of 30 kg/s. The dotted circles represent the pressure decline cone without anisotropy.

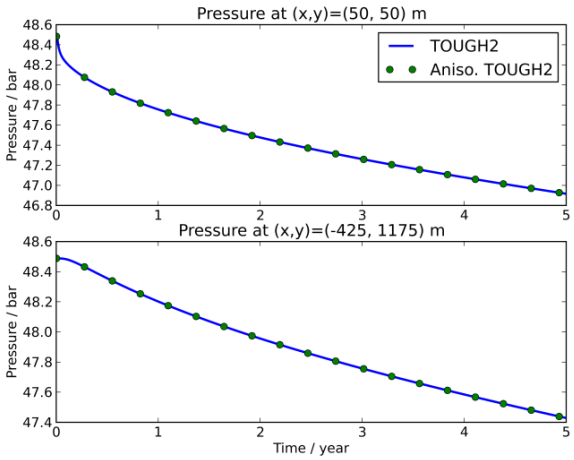

Figure 5. Comparison between the original and anisotropic methods in TOUGH2 for a system containing both liquid water and steam. The principal permeability axes are along x and y and the mesh rectangular to allow comparison between the two different methods.

Figure 4 shows the simulated pressure contours after five days of production. Figure 5 shows the comparison of pressure at two different

calibration points within the mesh. The results produced by the two different methods show that the methods agree to at least three significant digits at all times.

Five Different Regions

The following example compares TOUGH2 with the single-phase flow program Aqua3D (http://www.vatnaskil.is/softwaredevelopment/aqua3d-). Aqua3D is based on a finite element approximation with basis function on a triangular mesh. Aqua3D can properly handle anisotropy. Great care was taken to make the two simulations (TOUGH2 and Aqua3D) as similar as possible. Some discrepancy is to be expected, since the physical properties calculated by the EOS1 module may not exactly correspond to the constant values assumed by Aqua3d at all times.

This example consisted of a model containing five regions, each with different physical rock properties. The mesh was divided into a background region plus four additional rectangular regions (Figure 6). A producing well was located at $(x,y) = (0,0)$ m and produced at a constant rate of 30 kg/s.

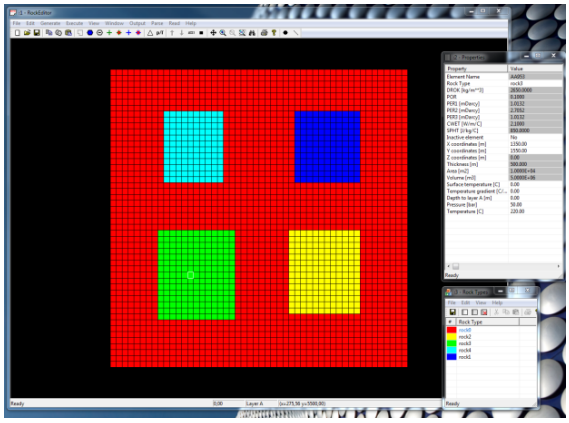

Figure 6. Hypothetical reservoir consisting of five different rock types.

The different rock properties were described by four parameters: porosity, permeability (along the primary principal axis), permeability ratio (between the two principal axes), and angle between the primary axis and the abscissa.

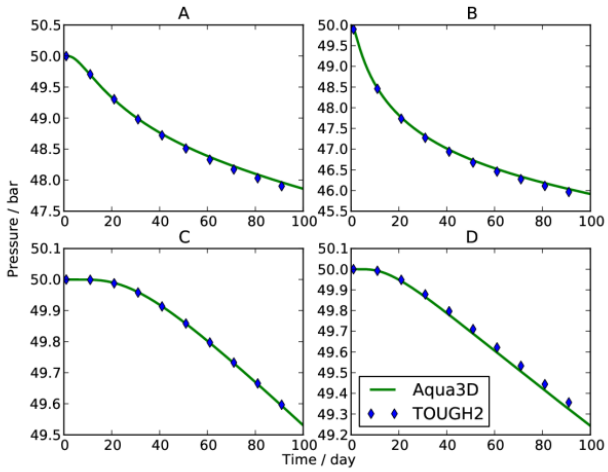

Figure 7. Comparison between TOUGH2 and Aqua3D for a model containing five different regions of anisotropic rock.

In each of the rectangular regions, one parameter differs from the background rock. The results of the simulations are presented in Figure 7 and Figure 8. Figure 7 shows the simulated pressure from TOUGH2 and Aqua3D at the four calibrations points.

Figure 8 shows simulated pressure contours at the end of the TOUGH2 run.

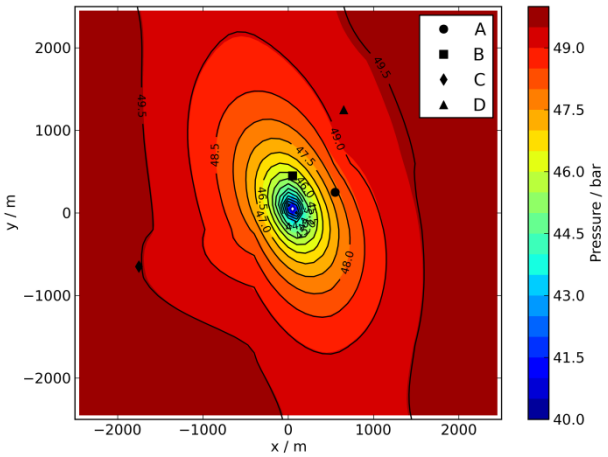

Figure 8. Areal plot of simulated pressure contours generated by TOUGH2 after 100 days. Black contour lines are Aqua3D results. The four symbols show the locations of the calculation points represented in Figure 7.

Although the methods employed in the two programs are different, Aqua3D and TOUGH2 gave similar results. This test validates our

approach in the case of nonhomogeneous anisotropic rock.

Multiple Layers

Previous test examples have all been single-layer systems. The following multilayer example was used to test whether the vertical flow (flow between horizontal layers), which includes the effects of gravity, has been correctly implemented into the new section of code handling anisotropy. As mentioned above, the solution used to implement anisotropy is a two-dimensional solution, and therefore the vertical flow is calculated in exactly the same way as in the original isotropic method. The vertical axis is therefore always a principal axis.

To allow for comparison between the original isotropic and the new anisotropic methods, the test example chosen is anisotropic but its principal axes are along *x*, *y*, and *z*. The same mesh as before was used, but the vertical depth was split into 10 layers of varying thickness (layers A–J). Initially, a run was performed without any sources or sinks to allow the pressures, under the effect of gravity, to reach equilibrium. Then, two sinks and one source were added to the system. The first sink (production rate of 50 kg/s) was added to an element in Layer C (Figure 9).

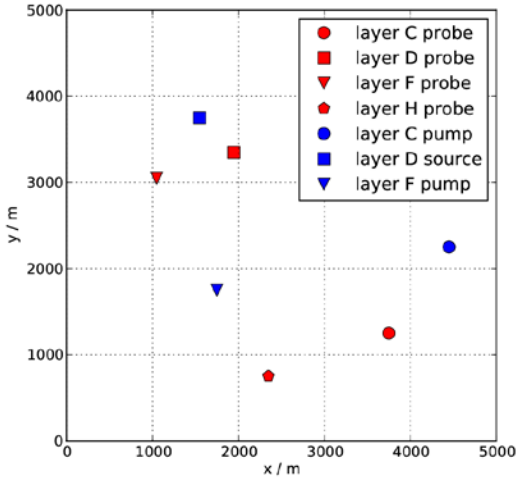

Figure 9. Top view of the multi-layer system showing the horizontal positions of the probes, sinks and source.

The second sink (production rate of 30 kg/s) was added to an element in Layer F. The source (rate of 15 kg/s) was added to an element in Layer D. Four measurement points were chosen to record the simulated pressure at different time intervals and produce a time series. Three of the measurement points were placed in the layers containing the sinks and the source (layers C, F and D), while the fourth probe was placed in layer H. A top view of the multilayer system in Figure 9 shows the horizontal positions of the probes, sinks, and source.

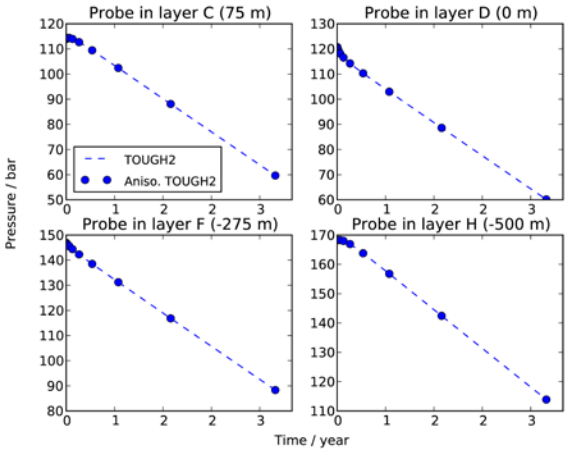

Figure 10. Simulated pressure time-series at each probe within the multi-layer system. Plot titles indicate the depth of the probes.

The simulated pressure time-series are shown in Figure 10 at each measurement point. The similarities in the curves produced by the isotropic and anisotropic methods indicate that the vertical flow between horizontal layers has been correctly implemented.

CONCLUSIONS

The result of this work is a fully functional addition to TOUGH2 and other simulators in the TOUGH family. This new addition includes the capability of defining anisotropic conditions within the model area. The code for the original isotropic method has been preserved, however, and it is therefore still possible to run a simulation using the unaltered version of TOUGH2. A switch was implemented to allow the user the option to change between the original isotropic method and the new anisotropic method. Input without any reference to anisotropy will automatically run using the original isotropic method. The new anisotropic method can only properly handle models where

the anisotropy is defined in a horizontal (*xy*) plane. Vertical flow is modeled using a scheme similar to the one used in the original isotropic method.

The new implementation has thoroughly tested with several examples and the results compared to either analytical solutions or other software packages. On all occasions, the comparison has been favorable.

Although not specifically dealt with in this report, it would be beneficial in the future to have the three new parameters available in iTOUGH2.

ACKNOWLEDGMENT

The authors acknowledge the cooperation with Karsten Pruess and Stefan Finsterle at LBNL.

REFERENCES

Abramowitz, M. and I. Stegun, *Handbook of mathematical functions: with formulas, graphs and mathematical tables*, Dover publications, USA, 1965.

Bear, J., *Hydraulics of groundwater*, McGraw-Hill Book Company, USA, 1979.

de Berg, M., M. van Krefeld, M. Overmans and O. Schwarzkopf, *Computational geometry: Algorithms and applications*, Springer, USA, 2000.

Grant, M., I. Donaldson and P. Bixley, *Geothermal reservoir engineering*, Academic Press Inc., USA, 1982.

Haukwa, C., *AMESH: A mesh creating program for the integral finite difference method: A User's Manual.*, Technical Report LBNL-45284, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1998.

Pruess, K., C. Oldenburg and G. Moridis, *TOUGH2 User's Guide, Version 2.0*, Report LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999.

Sigurðsson, S. and Vatnaskil Consulting Engineers , *Unpublished technical report*, 2009.

Strack, O., *Groundwater Mechanics*, Prentice-Hall Inc., New Jersey USA, 1989.